

Sliding Window Optimization on an Ambiguity-Clearness Graph for Multi-object Tracking

Qi Guo Le Dan Dong Yin Xiangyang Ji
Tsinghua University, Beijing, 100084, China

Abstract

Multi-object tracking remains challenging due to frequent occurrence of occlusions and outliers. In order to handle this problem, we propose an Approximation-Shrink Scheme for sequential optimization. This scheme is realized by introducing an Ambiguity-Clearness Graph to avoid conflicts and maintain sequence independent, as well as a sliding window optimization framework to constrain the size of state space and guarantee convergence. Based on this window-wise framework, the states of targets are clustered in a self-organizing manner. Moreover, we show that the traditional online and batch tracking methods can be embraced by the window-wise framework. Experiments indicate that with only a small window, the optimization performance can be much better than online methods and approach to batch methods.

1. Introduction

With the development of computer vision techniques, more and more people began to focus on understanding the behavior as well as other context of the objects via visual information. Tracking targets in video sequences, one of the core topics with wide applications in video surveillance, rocketed with the boost of tracking-by-detection (TBD) methods [26]. The TBD reconstruct the states of targets based on the detection responses by assigning identity to each detection and optimizing the trajectories [2, 21]. The prosperity of TBD these years has raised people's interests in a more challenging topic - multi-object tracking (MOT) with unknown numbers. MOT remains difficult due to complex settings of sequences, *e.g.*, intricate trajectories of targets, varying illumination, movements of cameras, *etc.*

The MOT problem can be handled in an online fashion, which could be adopted in time critical applications. However, the traditional online methods is susceptible to outliers brought by occlusions and noises, *e.g.*, false positives, true negatives, duplicate detections of a single target, *etc.* These outliers can cause ambiguities in data association. Some tackles the problem using sparse appearance model [19, 28],

and others via prediction [3] of states in future frames. But dynamics and appearances of the targets are unpredictable in some cases. Batch tracking methods are easier to solve the problem of outliers than online methods by global optimization of association and trajectories. Terms that penalize mutual exclusions and the number of tracklets [21, 9] were added to the energy function to regularize trajectories.

Apart from advantages of batch methods, one major problem is that the global optimization involves frames in the whole sequence [4] which does not suit for real-time applications. Some batch methods require initial solutions, *e.g.* [21]. Therefore, we propose our method in this paper, aiming at combining advantages of online and batch methods together while avoiding their disadvantages. We derive an iteratively Approximation-Shrink Scheme (AS Scheme) from the Maximum-A-Posterior (MAP) formulation using sequential approximation. We show that the state space can be effectively shrunk, but there may exist conflicts in the sequential optimization and the results may vary with different optimization sequences. In order to avoid these problems, an Ambiguity-Clearness Graph (A-C Graph) is formulated to efficiently represent the tracklet fragments and ambiguities in the association. A set of rules and procedures are defined for changes of nodes and edges in the graph, *e.g.*, connections, disconnections, transforms, merges, *etc.* A sliding Window-of-Ambiguity (WOA) is defined in the A-C Graph for sequential optimization of layers in the graph. Based on the A-C Graph and the sliding WOA optimization, MOT is conducted in a window-wise manner, which is able to disambiguate the association and accelerate the optimization process. We also show that the traditional online and batch approach can be embraced into this framework with different window sizes.

Our main contributions can be summarized as: (1) an approximation-shrink scheme that iteratively approximate the global optimization, (2) a window-wise optimization framework based on the novel A-C Graph which embrace the traditional online and batch methods, (3) a unified analysis of window-wise approaches with different window sizes using search tree.

2. Related Works

Different from the past tracking methods [24, 12], TBD reconstructs trajectories of targets by associating detections provided by the object detectors. Most of the researchers exploits the TBD framework to design their algorithms in MOT, which can be categorized as online and batch approaches.

As for batch tracking [21, 22, 2, 7, 25, 10, 8] approaches, conditional random field (CRF) is often used to learn and model the affinity such as appearance and motion to discriminate among different trajectories [29, 30]. A global and pairwise model is learned online in [30] to form an energy function, which is minimized offline via heuristic search. Despite the popularity of CRF model, extensive training is needed. Continuous energy model is introduced by a series of work [21, 22, 2]. Milan *et al.* [21] built a comprehensive continuous energy function by linearly combining terms regarding appearance, motion, mutual exclusion, trajectory persistence, *etc.* The continuous energy functions are easier to optimize than discrete ones, whereas they possess too many parameters and are hard to be tuned. Network flow is first applied to tracking by Zhang *et al.* [32]. A graph is formed with states of targets as nodes and the associations as edges. The likelihood of the states are represented as the capacity of edges. Butt *et al.* [7] improved the network structure by defining their node as a candidate pair of matching observations between consecutive frames. In order for a better model of occlusions, [25] designed a latent data association framework. Instead of assigning each detection to a corresponding track, they assume each detection is its own track and assign a latent data to each node to represent the association. In addition to the general modeling of targets, some people worked on tracking targets with specific characteristics, *e.g.*, Dicle *et al.* [10] focus on tracking targets with similar appearance but different motion patterns.

Online tracking [3, 4, 5, 6, 11, 31, 19] has become more and more popular these days. Network flow has also been adopted in online tracking. [5] formulate multi-object tracking into a multi-commodity network flow problem. They use sparse appearance to reduce computational complexity. Lu *et al.* [19] constructed a dictionary using already tracked objects and assigned the new detections by minimizing the L1 regularized function. Wang *et al.* [27] finds that the representation residuals follow the Laplacian distribution, by which they improved the sparse representation method on tracking. Hungarian algorithm is firstly introduced into tracking problems by Joo *et al.* [14] to solve the bipartite graph model they proposed. The frame-by-frame scheme of online tracking takes great advantages of hungarian algorithm. Bae *et al.* [3] designed tracklet confidence by considering the length, occlusion and affinity. Different strategies are applied to tracklets with high and low confidence. Hun-

garian algorithm is employed in the association for local and global association respectively. Hungarian algorithm greedily associates detections in consecutive frames which could possibly misses the global optimal and cause identity switches. Besides the popularity of Hungarian algorithm in association algorithms, Bayesian framework is also one of the most popular model for target modeling. Bae *et al.* [4] improved their previous work [3] by perform data association with a track existence probability, the provided detections are associated to the existed tracks and the corresponding track existence probabilities will be updated. Yoon *et al.* [31] constructed a Relative Motion Network(RMN) to factor out the camera motion by considering motion context from multiple object and incorporate relative motion network to Bayesian framework.

3. Approximate-Shrink Scheme

Given observations $\mathbb{Z}_{1:t} = \{Z_{(\tau,i)} | \tau = 1, 2, \dots, t, i = 1, 2, \dots, n_\tau\}$ of a real time video sequence, where n_τ denotes the number of observations in frame τ , we assume: (1) each observation $Z_{(\tau,i)}$ corresponds to a state $X_{(\tau,i)}$ [25], (2) states in the same frame are independent, (3) some of the states are already clear given observations. The Maximum-a-Posterior (MAP) formulation of MOT is

$$\hat{\mathbb{X}}_{1:t} = \arg \max_{\mathbb{X}_{1:t}} P(\mathbb{X}_{1:t} | \mathbb{Z}_{1:t}). \quad (1)$$

Based on Assumption (2), we resolve $\mathbb{X}_{1:t}$ as

$$\hat{\mathbb{X}}_{1:t} = \arg \max_{\mathbb{X}_{1:t}} \prod_{\tau=1}^t \prod_{i=1}^{n_\tau} P(X_{(\tau,i)} | \mathbb{Z}_{1:t}, \mathbb{X}_{1:\tau-1}). \quad (2)$$

Assumption (3) offers us an intuition that there exist some states $\mathbb{X}_{1:t}^C = \{X_{(\tau',j)} | P(X_{(\tau',j)} | \mathbb{Z}_{1:t}, \mathbb{X}_{1:\tau'-1}) \approx P(X_{(\tau',j)} | \mathbb{Z}_{1:t})\}$. Denote $\mathbb{X}_{1:t}^A = \mathbb{X}_{1:t} \setminus \mathbb{X}_{1:t}^C$. We name $\mathbb{X}_{1:t}^C$ Clear states (C states) and $\mathbb{X}_{1:t}^A$ Ambiguous states (A states). The global optimization in Equation 2 can be relaxed to

$$\hat{\mathbb{X}}_{1:t}^A = \arg \max_{\mathbb{X}_{1:t}^A} \prod_{\forall X_{(\tau,i)} \in \mathbb{X}_{1:t}^A} P(X_{(\tau,i)} | \mathbb{Z}_{1:t}, \mathbb{X}_{1:\tau-1}), \quad (3)$$

and

$$\hat{X}_{(\tau',j)} = \arg \max_{X_{(\tau',j)}} P(X_{(\tau',j)} | \mathbb{Z}_{1:t}, \forall X_{(\tau',j)} \in \mathbb{X}_{1:t}^C). \quad (4)$$

Doing these two optimization separately is an approximation to Equation 2. First, we sequentially optimize every state $X_{(\tau',j)}$ in $\mathbb{X}_{1:t}^C$ (approximation step) via Equation 4. Then we set $\mathbb{X}_{1:t}^C$ fixed as the evidence for $\mathbb{X}_{1:t}^A$, and derive Equation 3 to

$$\hat{\mathbb{X}}_{1:t}^A = \arg \max_{\mathbb{X}_{1:t}^A} \prod_{\forall X_{(\tau,i)} \in \mathbb{X}_{1:t}^A} P(X_{(\tau,i)} | \mathbb{Z}_{1:t}, \mathbb{X}_{1:\tau-1}^A, \mathbb{X}_{1:t}^C) \quad (5)$$

(shrink step). We iteratively find the $\mathbb{X}_{1:t}^{C'} \in \mathbb{X}_{1:t}^A$, $\mathbb{X}_{1:t}^{C'} = \{X_{(\tau',j)} | P(X_{(\tau',j)} | \mathbb{Z}_{1:t}, \mathbb{X}_{1:\tau'-1}^A, \mathbb{X}_{1:t}^{C'}) \approx P(X_{(\tau',j)} | \mathbb{Z}_{1:t}, \mathbb{X}_{1:t}^{C'})\}$, let $\mathbb{X}_{1:t}^{C'} = \mathbb{X}_{1:t}^{C'}$ and repeat the above steps to shrink the search space.

This Approximate-Shrink Scheme (A-S Scheme) iteratively search and narrow down the state space. $\mathbb{X}_{1:t}^C$ serve as nucleus of trajectories in the space which attract states to associate to them. Some nucleus merge together in the iteration to form longer tracklets during the iteration. However, the space is still too large, and the convergence is not guaranteed. More approximations are needed to accelerate the speed and ensure the convergence of this scheme. Moreover, it is necessary to design a data structure so as to avoid conflicts of associations of states in $\mathbb{X}_{1:t}^C$ and the effects of the sequence on the optimization results. Therefore, we propose a self-organizing A-C Graph and window-wise optimization framework to meet the demands in this regard.

4. Window-wise Optimization for Tracking

4.1. Ambiguous-Clearness Graph

Given states $\mathbb{X} = \{X_{(\tau,i)} | \tau = 1, 2, \dots, t, i = 1, 2, \dots, n_\tau\}$ and observations $\mathbb{Z} = \{Z_{(\tau,i)} | \tau = 1, 2, \dots, t, i = 1, 2, \dots, n_\tau\}$ (the detections serve as observations in TBD multi-object tracking) in a real time video sequence, predefined thresholds C_{thre} and A_{thre} (the value of C_{thre} and A_{thre} are given in Section 5), we define state $X_{(\tau',j)}$ to be the *parent* of state $X_{(\tau,i)}$ if $\tau' < \tau$ and there exists an association between $X_{(\tau',j)}$ and $X_{(\tau,i)}$, and $X_{(\tau,i)}$ is the *child* of $X_{(\tau',j)}$. ($X_{(\tau,i)}$ and $X_{(\tau',j)}$ are only used as examples for clearness in illustration. They do not indicate certain states.) The *determined parent* of a state is its only parent and the affinity score of the association is greater than C_{thre} . We now formally define the C states and A states. If a state $X_{(\tau,i)}$ has one determined parent or does not have parent, $X_{(\tau,i)}$ is a *clear state* (C state), denoted as $X_{(\tau,i)}^C$. On the contrary, if $X_{(\tau,i)}$ has parent states but does not have a determined parent, it is an *Ambiguous State* (A state), denoted as $X_{(\tau,i)}^A$. Note that a C state can only have zero or one parent. All the parents of a state $X_{(\tau,i)}$ form its *active set*. We regulate that a state can have up to one C state as its child, and the frame number of its A state child should be smaller than that of its C state child. The observation corresponding to $X_{(\tau,i)}^C$ and $X_{(\tau,i)}^A$ is notated as $Z_{(\tau,i)}^C$ and $Z_{(\tau,i)}^A$. A *clear association* is the association between a clear state and its parent, and a *tracklet* is defined as a group states connected by clear association. The tracklet including $X_{(\tau,i)}$ is denoted as $Trk(X_{(\tau,i)})$. The C states in $Trk(X_{(\tau,i)})$ after $X_{(\tau,i)}$ is defined as the *descendant* of $X_{(\tau,i)}$. By taking states and associations as the vertices and edges, we form the A-C Graph of the MOT problem. In this paper, we use states and associations instead of vertices and edges when discussing on the A-C Graph. The A-C Graph

Functions and Symbols	Description
$isempty(StateSet)$	Check whether the <i>StateSet</i> is empty.
$find(StateSet = CState)$	Find the C States in the <i>StateSet</i> .
$X_{(\tau,i)}.pStates$	Find all the parents of $X_{(\tau,i)}$.
$X_{(\tau,i)}.cldStates$	Find all the children of $X_{(\tau,i)}$.
$X_{(\tau,i)}.frameN$	Find the frame number of $X_{(\tau,i)}$.
$X_{(\tau,i)}.isClear$	Judge whether $X_{(\tau,i)}$ is a clear state.
$X_{(\tau,i)}.conf$	The affinity scores between all the fathers of $X_{(\tau,i)}$ and $X_{(\tau,i)}$.

Table 1. The functions and symbols used in this paper.

of TUD-Stadtmitte dataset is visualized in Figure 4, where the clear association is shown in solid line and the states belong to the same tracklet is in the same color.

As the association is directed from parent to child, the A-C Graph is a directed acyclic graph. In an A-C Graph, we define a time period τ' to τ ($1 \leq \tau' < \tau \leq t$) where there is only clear association in 1 to $\tau' - 1$ and $\tau + 1$ to t as Window-of-Ambiguity (WOA). The tracklet outside the WOA is determined and fixed and the changes of the states and association can only take place in the WOA. One can restrict the size of state space by setting the length of WOA.

4.2. Actions

As is mentioned in Section 3, actions in A-C Graph should help avoid conflicts, *e.g.*, multiple fathers for a C state, multiple C state children, clear association forms cycle, *etc.* Meanwhile, the actions should be symmetrical to avoid the effect of chronological order. The basic actions of A-C Graph are initializations, disconnections, connections and merges between two states. Table 1 shows functions and symbols used in defining these actions.

For a newly-entered state $X_{(\tau,i)}$, first we initialize the active set by enumerating all the potential parents. As is regulated in Section 4.1, $X_{(\tau,i)}$ is able to connect with states in the previous frames, who does not have C state child or whose C state child is after $X_{(\tau,i)}$. Procedure 1 shows the pseudocode of initializing the active set.

We disconnect two states $X_{(\tau,i)}$ and $X_{(\tau',j)}$ by removing the association between them, and update these two states.

As is shown in Procedure 2, we assign $X_{(\tau,i)}$ to $X_{(\tau',j)}$ as A state child. The procedure is terminated if $X_{(\tau,i)}$ is already a C state. If not, we check the descendant of $X_{(\tau',j)}$. If $X_{(\tau',j)}$ has no descendants, we directly add an association between $X_{(\tau,i)}$ and $X_{(\tau',j)}$, otherwise, we find the nearest C state descendant X^p in the tracklet of $X_{(\tau',j)}$ not after $X_{(\tau,i)}$. If X^p is in frame τ , the procedure is terminated. If X^p is before $X_{(\tau,i)}$, add the association between $X_{(\tau,i)}$ and X^p .

Procedure 5 illustrates the action that $X_{(\tau,i)}$ is connected to $X_{(\tau',j)}$ as C state child. If $X_{(\tau,i)}$ is currently not a C state, the existing parents of $X_{(\tau,i)}$ are removed. If $X_{(\tau',j)}$ does not have C state children, we directly add a connection between $X_{(\tau,i)}$ and $X_{(\tau',j)}$, otherwise, we find $X_{(\tau',j)}$'s latest

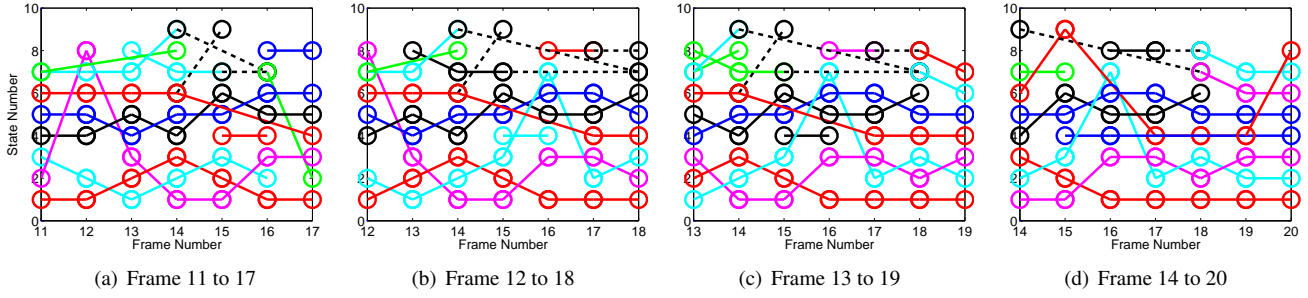


Figure 1. Visualization of the Window of Ambiguity (WOA) from frame 12 to frame 21 in TUD-Stadtmitte dataset. Each association is directed from parent to child and the A-C Graph is directed and acyclic. From (a) to (b), state $X_{(16,7)}$ was connected to state $X_{(14,2)}$ as a C state child, and was merged with $X_{(16,2)}$. Meanwhile, the association from $X_{(14,9)}$ and $X_{(14,7)}$ to $X_{(16,7)}$ were removed. From (c) to (d), state $X_{(15,9)}$ was inserted into the tracklet of state $X_{(14,6)}$ and state $X_{(17,4)}$. The figure is best shown in color.

C state descendant X^p not after $X_{(\tau,i)}$. If X^p is in frame τ , X^p and $X_{(\tau,i)}$ are merged together via Procedure 6. As X^p is before $X_{(\tau,i)}$, an association is added between $X_{(\tau,i)}$ and X^p . All the A and C state children of X^p after $X_{(\tau,i)}$ are removed from X^p and reconnected to $X_{(\tau,i)}$ following Procedure 2 and 5 respectively. If $X_{(\tau,i)}$ is currently a C state and $X_{(\tau',j)}$ is not, $X_{(\tau',j)}$ is inserted into $X_{(\tau,i)}$'s tracklet using Procedure 5 if there is not a state in frame τ' in $X_{(\tau,i)}$'s tracklet and Procedure 6 if there exists a state X^{p1} in frame τ' . If $X_{(\tau,i)}$ and $X_{(\tau',j)}$ are both C states, the two tracklets $X_{(\tau,i)}$ and $X_{(\tau',j)}$ will be grouped into one by recursively calling Procedure 5 and 6, as shown in Procedure 5. If one of the two states is in a tracklet, the other state will be inserted into the tracklet.

Procedure 6 describes the process of merging $X_{(\tau,j)}$ to $X_{(\tau,i)}$ in the same frame. As we cannot make changes on the states and tracklets outside WOA, we ensure that $X_{(\tau,j)}$ and $X_{(\tau,i)}$ cannot be C states at the same time to avoid merging of states outside WOA. For the descendants of $X_{(\tau,i)}$ and $X_{(\tau,j)}$, we recursively merge them into one tracklet by Procedure 5. For the A state child X^{cld} of $X_{(\tau,j)}$, we simply remove the association between X^{cld} and $X_{(\tau,j)}$ and connect it to $X_{(\tau,i)}$ via Procedure 2.

Procedure 1 Initialize the active set for the state $X_{(\tau,i)}$.

Input: state $X_{(\tau,i)}$, latest frame number t , size of Window of Ambiguity (WOA) l

Output: the active set containing all the potential parents of $X_{(\tau,i)}$

for all the states $X_{(\tau',j)}$ in frame $t - l + 1$ to τ **do**
 if $isempty(find(X_{(\tau',j)}.cldStates = CState))$ **or**
 $find(X_{(\tau',j)}.cldStates = CState).frameN \leq \tau$ **then**
 Add $X_{(\tau',j)}$ to the active set
 end if
end for

Although there exists recursion in the actions, it can be easily proved that the recursion in Procedure 2, 5 and 6 cannot form an endless recursion loop, and the sequence of carrying out actions on a set of states will not affect the

Procedure 2 Connect state $X_{(\tau,i)}$ to state $X_{(\tau',j)}$ as A state child.

Input: child state $X_{(\tau,i)}$, parent state $X_{(\tau',j)}$

Output: the updated network

if $X_{(\tau,i)}.isClear = false$ **then**
 $X^p = X_{(\tau',j)}$
 while (**not** $isempty(find(X^p.cldStates = CState))$) **and**
 $X^p.frameN \leq \tau$ **do**
 X^p = the C state child of X^p
 end while
 if $X^p.frameN < \tau$ **then**
 Add $X_{(\tau,i)}$ to $X^p.cldStates$
 Add X^p to $X_{(\tau,i)}.pStates$
 Update the features of $X_{(\tau,i)}$ and X^p
 end if
end if

Procedure 3 Connect state $X_{(\tau,i)}$ to state $X_{(\tau',j)}$ as C state child, where $X_{(\tau,i)}$ is currently an A state.

Input: child state $X_{(\tau,i)}$, parent state $X_{(\tau',j)}$, latest frame number t , size of Window of Ambiguity (WOA) l

Output: the updated network

$X^p = X_{(\tau',j)}$
while (**not** $isempty(find(X^p.cldStates = CState))$) **and**
 $X^p.frameN \leq \tau$ **do**
 X^p = the C state child of X^p
end while
if $X^p.frameN = \tau$ **then**
 Do Procedure 6 with $(X_{(\tau,i)}, X^p, t, l)$ as input
else
 Remove all parents of $X_{(\tau,i)}$
 Remove all children of X^p in the same frame with $X_{(\tau,i)}$
 for all children X^{cld} of X^p in the frames after $X_{(\tau,i)}$ **do**
 Remove the association between X^{cld} and X^p
 if $X^{cld}.isClear = true$ **then**
 Do Procedure 5 with $(X^{cld}, X_{(\tau,i)}, t, l)$ as input
 else
 Do Procedure 2 with $(X^{cld}, X_{(\tau,i)})$ as input
 end if
 end for
 end if
end if

Procedure 4 Connect state $X_{(\tau,i)}$ to state $X_{(\tau',j)}$ as C state child, where $X_{(\tau,i)}$ is currently a C state.

Input: child state $X_{(\tau,i)}$, parent state $X_{(\tau',j)}$, latest frame number t , size of Window of Ambiguity (WOA) l
Output: the updated network

```

 $X^{p1} = X_{(\tau',j)}$ 
while ( $X^{p1}.isClear = true$ ) and  $X^{p1}.frameN > t - l$  do
   $X^{p1} =$  the determined father of  $X^{p1}$ 
end while
 $X^{p2} = X_{(\tau',j)}$ 
while ( $X^{p2}.isClear = true$ ) and  $X^{p2}.frameN > t - l$  do
   $X^{p2} =$  the determined father of  $X^{p2}$ 
end while
if  $X^{p1}.frameN > X^{p2}.frameN$  then
  Do Procedure 5 with ( $X^{p2}, X^{p1}, t, l$ ) as input
else if  $X^{p2}.frameN > X^{p1}.frameN$  then
  Do Procedure 5 with ( $X^{p1}, X^{p2}, t, l$ ) as input
else
  if  $X^{p1}.isClear = true$  and  $X^{p2}.isClear = true$  then
    if  $X^{p1}.conf \geq X^{p2}.conf$  then
      Remove the parent of  $X^{p2}$ ,  $X^{p2} = AState$ 
      Do Procedure 6 with ( $X^{p1}, X^{p2}, t, l$ ) as input
    else
      Remove the parent of  $X^{p1}$ ,  $X^{p1} = AState$ 
      Do Procedure 6 with ( $X^{p2}, X^{p1}, t, l$ ) as input
    end if
  else if  $X^{p1}.isClear = true$  then
    Do Procedure 6 with ( $X^{p1}, X^{p2}, t, l$ ) as input
  else
    Do Procedure 6 with ( $X^{p2}, X^{p1}, t, l$ ) as input
  end if
end if

```

Procedure 5 Connect state $X_{(\tau,i)}$ to state $X_{(\tau',j)}$ as C state child.

Input: child state $X_{(\tau,i)}$, parent state $X_{(\tau',j)}$, latest frame number t , size of Window of Ambiguity (WOA) l
Output: the updated network

```

if  $X_{(\tau,i)}.isClear = false$  then
  Do Procedure 3 with ( $X_{(\tau,i)}, X_{(\tau',j)}, t, l$ )
else
  Do Procedure 4 with ( $X_{(\tau,i)}, X_{(\tau',j)}, t, l$ )
end if

```

structure of A-C Graph. Visualization of these actions in TUD-Stadtmitte dataset can be found in Figure 4. In Figure 1(b), newly-entered states $X_{(18,1)}$ to $X_{(18,8)}$ connected to their initial active sets via Procedure 1, 2 and 5. From Figure 1(a) to 1(b), $X_{(16,7)}$ was connected to $X_{(14,2)}$ as a C state child by Procedure 5, and merged with $X_{(16,2)}$ using Procedure 6.

4.3. Sliding Window Optimization

For a real time sequence, the A-C Graph is continuously adding new states from latest frame t . The WOA should be sliding to keep its size from being too large and remove the ambiguities to generate tracks. So we set the upper bound of the size of WOA as l .

The sliding window optimization consists of three steps.

Procedure 6 Merge state $X_{(\tau,j)}$ with state $X_{(\tau,i)}$.

Input: state $X_{(\tau,i)}$, state $X_{(\tau,j)}$, latest frame number t , size of Window of Ambiguity (WOA) l
Output: the updated network

```

if  $X_{(\tau,j)}.isClear$  then
  Remove  $X_{(\tau,j)}$  from its parent  $X^p$ , if any
  Do Procedure 5 with ( $X_{(\tau,i)}, X^p, t, l$ ) as input
else
  Remove  $X_{\tau}$  from its parents  $X^p$ 
  For all  $X^p$ , do Procedure 2 with ( $X_{(\tau,i)}, X^p$ ) as input
end if
for all the children  $X^{cld}$  of  $X_{(\tau,j)}$  do
  if  $X^{cld}.isClear = true$  then
    Remove  $X^{cld}$  from  $X_{(\tau,j)}$ 
    Do Procedure 5 with ( $X^{cld}, X_{(\tau,i)}, t, l$ ) as input
  else
    Remove  $X^{cld}$  from  $X_{(\tau,j)}$ 
    Do Procedure 2 with ( $X^{cld}, X_{(\tau,i)}, t, l$ ) as input
  end if
end for

```

First, for all the newly-entered states $X_{(t,i)}$ in frame t , $i = 1, \dots, n_t$, we find the active sets via Procedure 1 and compute the affinity score $a(X_{(t,i)}, X^p)$ between $X_{(t,i)}$ and each state X^p in the corresponding active set. If $a(X_{(t,i)}, X^p) \geq C_{thre}$, do Procedure 5 with ($X_{(t,i)}, X^p, t, l$) as input. If $A_{thre} < a(X_{(t,i)}, X^p) < C_{thre}$, do Procedure 2 with ($X_{(t,i)}, X^p$) as input. Second, from frame $t - l$ to t , we sequentially recompute the affinity score of states in the same frame with their fathers and reconnect them according to the new affinity. Third, Hungarian Algorithm [1] is carried out on states in frame $t - l$ with their father states to get the best arrangement of association and clear all the ambiguity in frame $t - l$. All states in frame $t - l$ are transformed to C states and the WOA shifts forward. If t has not reached the end, $t = t + 1$ and return to the first step, otherwise, $l = l - 1$ and redo the third step. The outline of the optimization process is shown in Procedure 7.

Procedure 7 Conduct sliding window optimization for MOT.

Input: size of Window-of-Ambiguity (WOA) l
Output: the final A-C Graph and the association result

1. Associate the newly-entered states in the latest frame t to their initial active sets.
2. Sequentially shrink the active set of each A state in WOA.
3. Determine the association of states in frame $t - l$ using Hungarian Algorithm [1].

```

if  $t$  has not reached the end then
   $t = t + 1$ , return to 1
else
   $l = l + 1$ , return to 3
end if

```

The sliding window optimization conducts A-S Scheme in a window-wise manner. Procedure 5 and 6 in step one and two serve as the approximation step, and updating affin-

ity score in step two follows the shrink step. Step three forces the states in frame $t - l$ to determine their connections, which guarantees the convergence.

4.4. Online, Delayed and Batch Methods

Based on the definition of A-C Graph and sliding window optimization, we form this window-wise framework which includes online ($l = 1$), delay ($1 < l < t$) and batch methods ($l = t$). Figure 2 demonstrates the formation of a trajectory starting from X^s in the A-C Graph via these three methods. The window-wise optimization finds a relatively small search tree T_1, \dots, T_{t-l+1} according to l at each iteration. As for an online method (Figure 2(b)), $l = 1$ and the search is greedy. For a delayed method (Figure 2(c)), heuristic search is conducted in T_1, \dots, T_{t-l+1} . The search space remains unchanged for a batch method (Figure 2(d)), so local search methods, *e.g.*, hill climbing, simulated annealing, *etc.*, is often exploited to direct to local optimal iteratively. The experimental analysis of the relation between l and optimization results is provided in Section 5.2.

5. Experimental Evaluation

5.1. Implementation

Affinity model: We implemented a basic affinity model, following [3], which includes the appearance model $App(X_{(\tau,i)}, X_{(\tau',j)})$, motion model $Mot(X_{(\tau,i)}, X_{(\tau',j)})$ and shape model $Shp(X_{(\tau,i)}, X_{(\tau',j)})$. The appearance model measures the Bhattacharyya distance of histograms of $X_{(\tau,i)}$ and $X_{(\tau',j)}$. If $X_{(\tau,i)}$ is in a tracklet $Trk(X_{(\tau,i)})$, instead of using Incremental Linear Discriminant Analysis (ILDA) used in [3], we simply average the appearance histograms of all states in $Trk(X_{(\tau,i)})$ using an exponential discount factor. First-order Kalman filter is applied to smoothing and predicting positions of the targets and shapes of the bounding boxes. We compute the normalized distance of target positions and bounding box shapes and map them to a Gaussian distribution $N(O, Var)$ to get the affinity scores. The overall affinity

$$Aff(X_{(\tau,i)}, X_{(\tau',j)}) = App \times Mot \times Shp. \quad (6)$$

Dataset description: We use the MOT Benchmark [18] for training and evaluation in this paper, where the benchmark contains both 11 sequences for training and testing. In total, there are 11, 286 frames, 5, 503 for training set and 5, 783 for testing set. The sequences possess different frame rates and resolutions, and only tracking pedestrians.

Parameter Settings: In our experiment, the $C_{thre} = 0.5$ and $A_{thre} = 0.1$. We estimate the length of every occlusion (number of frames with overlap > 0.4) in the training set of MOT Benchmark and study the distribution of occlusion lengths. As shown in Figure 3, about 99% of the overlaps are within 5s, and 84% of which are within 1s. Therefore,

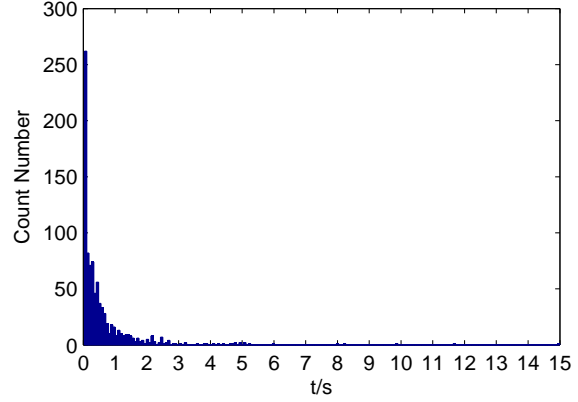


Figure 3. Distribution of lengths of bounding box overlaps in the ground truth sequences in MOT Benchmark [18]. 99% of the overlaps are within 5s and 84% of them are in 1s.

the delayed time is set to 1s and the length of WOA $l = \text{frame rate} \times \text{delayed time}$. The variance of the Gaussian distribution in the motion model and shape model is $Var = [20^2, 50^2]$. Other parameters of the affinity model are the same as [3].

5.2. Analysis of Window-of-Ambiguity

To analyze the connection of WOA size and the quality of the window-wise optimization, we define the energy of an A-C Graph as

$$E(t) = - \sum_{\tau=1}^t \sum_{i=1}^{n_{\tau}} \max_{X^p \in X_{(\tau,i)} \cdot pStates} Aff(X_{\tau,i}, X^p). \quad (7)$$

Figure 4 presents the final energy with varying size of WOA on TUD-Stadtmitte (number of frame = 179), TUD-Campus (number of frame = 71) and PETS-S2L2 (number of frame = 436) in MOT Benchmark. The X-axis is in logarithmic scale. Interestingly, final energy of these sequences reduced significantly when window size l grows from 1 to 5, while the speed of decrease become much slower when $l > 5$. Settings of these sequences, *e.g.*, target density, viewpoint, *etc.*, are different, but the patterns of energy change almost remain identical. It is likely that the trend of final energy only deals with WOA size l . And the tracking results can be much improved with a small WOA comparing to the online method, which experimentally illustrates the better performance of delayed methods than online ones in the window-wise optimization framework. The final energy does not reduce too much when l grows larger than 5. This indicates the sliding window approximation only has a minor effect on the final performance. And it becomes a trade-off between speed and better results when WOA grows larger.

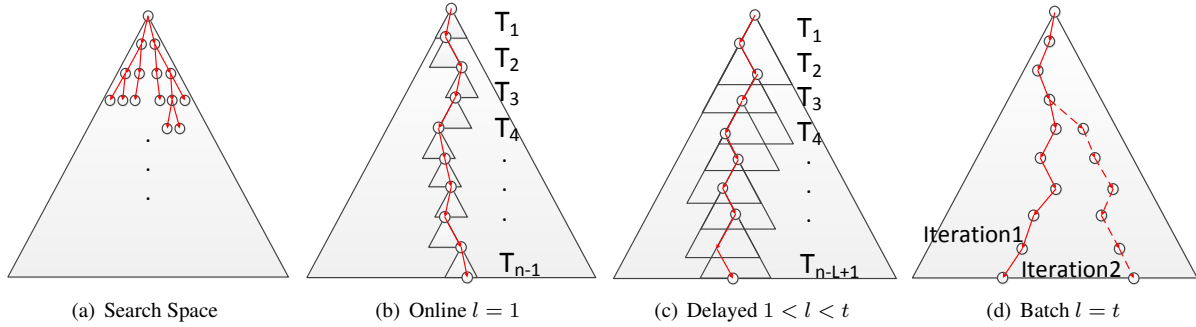


Figure 2. Formation of a trajectory with different l . (a) illustrates the original search space. (b), (c) and (d) stand for the search process with local search tree. T_τ indicates the SWO in frame τ , $\tau = 1, \dots, (t - l + 1)$. The red lines represent the associations. For online and delayed approaches, the trajectory are formed from top to bottom, while for batch approaches, the trajectory is formed and optimized via iteration.

Method	AC-MOT	CEM [21]	MotiCon [17]	SegTrack [20]
type	Delayed	Batch	Batch	Batch
TUD-Crossing	62.3	61.6	58.2	53.9
ETH-Linthescher	18.2	18.4	18.3	11.1
ETH-Crossing	23.4	18.2	22.8	23.4
KITTI-16	38.1	31.6	38.8	40.2

Table 3. MOTA of some sequences in MOT Benchmark. We compare our method using [3]’s affinity model with state-of-the-art affinity models.

5.3. Performance Evaluation

Evaluation Metrics: We apply the CLEAR MOT [15] and [29, 16]’s metric when evaluating our result. The multiple object tracking accuracy (MOTA) shows the combined accuracy based on the number of false positives (FP), identity switches (IDS) and missed targets (FN). The multiple object tracking precision (MOTP) measures the overlap of bounding boxes between ground truths and results given by trackers. MT and ML indicate the number of mostly tracked and lost targets. FG represents the number of fragmented tracks.

Evaluation: As shown in Table 5.3, our method clearly outperforms the TC.ODAL method using the same affinity model, not only in MOTA. Even in some datasets, shown in Table 5.3, our method with the basic affinity model reached the performance of the methods using state-of-the-art affinity models.

6. Conclusion

This paper proposed an A-S Scheme for sequential approximation and a window-wise optimization framework based on the A-C Graph. The core idea of this method is to cluster the states subject to several constraints, *e.g.* states in the same frame cannot be clustered into one group, *etc.* The A-C Graph together with the sliding window optimization transformed the global clustering into a sequen-

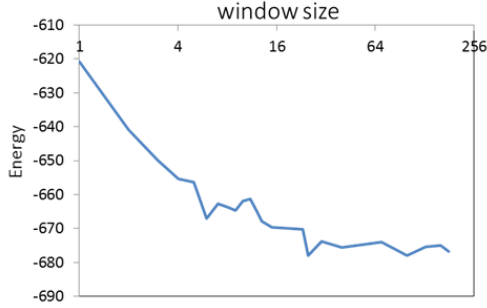
tial local clustering which self-organized the structure in a relatively small state space, which can be done efficiently with little harm to handling occlusions. We showed experimentally that the characteristics of window-wise optimization framework rarely change with the varying settings of the sequence. As the affinity model serves as the distance metric in clustering, it can influence the results of clustering. Therefore, it is a fair comparison of optimization models if similar affinity models are adopted. The experimental results show that by using the basic affinity model, our method even showed competitive performance in an unfair test. Our future work is to realize more state-of-the-art affinity models to the window-wise optimization model. Also, we plan to design a unity interface, which can help to embed the affinity models into different optimization models much easier than now.

References

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows*. Prentice Hall, 1993.
- [2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1926–1933. IEEE.
- [3] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1218–1225. IEEE.
- [4] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking with data association and track management. 2014.
- [5] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Multi-commodity network flow for tracking multiple people. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1614–1627, 2014.
- [6] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and*

Method	Type	MOTA \uparrow	MOTP \uparrow	MT \uparrow	ML \downarrow	FP \downarrow	FN \downarrow	IDS \downarrow	FG \downarrow
AC-MOT(Proposed + affinity of [3] \dagger)	Delayed	18.1 \pm 17.9	70.4	5.8%	58.3%	13,492	36,295	509	1,092
TBD[13]	Batch	15.9 \pm 17.6	70.9	6.4%	47.9%	14,943	34,777	1,939	1,963
TC-ODAL [3] \dagger	Online	15.1 \pm 15.0	70.5	3.2%	55.8%	12,970	38,538	637	1,716
DP.NMS [23]	Batch	14.5 \pm 13.9	70.8	2.3%	40.8%	13,171	34,814	4,537	3,090

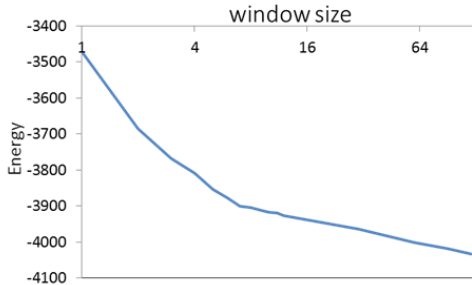
Table 2. Performance evaluation. Results can be found in http://motchallenge.net/results/2D_MOT_2015/. The best outcomes are marked in bold. \uparrow represents higher is better, while \downarrow stands for lower being better. Methods evaluated using the same set of affinity descriptor are marked with the same symbol.



(a) TUD-Stadtmitte(number of frame = 179)



(b) TUD-Campus(number of frame = 71)



(c) PETS-S2L2(number of frame = 436)

Figure 4. The final energy with varying size of Window-of-Ambiguity (WOA) on different sequences. The X-axis is in logarithmic-scale. The energy decrease rapidly when l grows from 1 to 5. When $l > 5$, the decrease of energy becomes slower.

Machine Intelligence, IEEE Transactions on, 33(9):1820–1833, 2011.

- [7] A. A. Butt and R. T. Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Computer*

Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 1846–1853. IEEE.

- [8] C. Canton-Ferrer, J. R. Casas, M. Pardis, and E. Monte. Multi-camera multi-object voxel-based monte carlo 3d tracking strategies. *EURASIP Journal on Advances in Signal Processing*, 2011(1):1–15, 2011. B and limited O 3D.
- [9] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. *arXiv preprint arXiv:1504.02340*, 2015.
- [10] C. Dicle, O. I. Camps, and M. Sznaiar. The way they move: Tracking multiple targets with similar appearance. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2304–2311. IEEE.
- [11] C. Fantacci, B.-N. Vo, B.-T. Vo, G. Battistelli, and L. Chisci. Consensus labeled random finite set filtering for distributed multi-object tracking. *arXiv preprint arXiv:1501.01579*, 2015. new approach.
- [12] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, 1983. B.
- [13] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(5):1012–1025, 2014.
- [14] S.-W. Joo and R. Chellappa. A multiple-hypothesis approach for multiobject visual tracking. *Image Processing, IEEE Transactions on*, 16(11):2849–2854, 2007.
- [15] B. Keni and S. Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.
- [16] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1217–1224. IEEE. B.
- [17] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3542–3549. IEEE, 2014.
- [18] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.
- [19] W. Lu, C. Bai, K. Kpalma, and J. Ronsin. Multi-object tracking using sparse representation. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 2312–2316. IEEE. O.

- [20] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets.
- [21] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(1):58–72, 2014.
- [22] A. Milan, K. Schindler, and S. Roth. Detection-and trajectory-level exclusion in multiple object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3682–3689. IEEE.
- [23] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.
- [24] D. B. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979. B.
- [25] A. V. Segal and I. Reid. Latent data association: Bayesian model selection for multi-target tracking. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2904–2911. IEEE.
- [26] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468, 2014.
- [27] B. Wang and F. Wang. Multi-object tracking using least absolute deviation. In *Image and Signal Processing (CISP), 2014 7th International Congress on*, pages 60–65. IEEE. O.
- [28] D. Wang, H. Lu, and M.-H. Yang. Least soft-threshold squares tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2371–2378. IEEE. O single.
- [29] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2034–2041. IEEE.
- [30] B. Yang and R. Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. *International Journal of Computer Vision*, 107(2):203–217, 2014.
- [31] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 33–40. IEEE. O.
- [32] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.